

PYTHON FUNCTION

- A function is a block of code which only runs when it is called.
(C ફુંક્શન એવી કોડની રીત બોલ કી જે હજુ વ્યાપ્ત નથી કે જ્યારે તૈનિ વીઠાવામાં આવી છે.)
- You can pass data, known as parameter, into a function.
(C તમે ફુંક્શનમાં કોઈ પાસ કરી શકો હો, જેની પરિણામાં મળું આવે છે.)
- A function can return data as a result.
(ફુંક્શન રાખી કોઈ પદ્ધતિ કરી શકે હો.)

*→ Calling a Function

To call a function, use the function name followed by parenthesis.

(કોઈ ફુંક્શન ની કોઈ કઢી હોય, ત્યારે કોઈ પણ ફુંક્શન નામ ની ઉપર્યુગીંગ કરીએ આવી છે.)

```
def my_function():
    print("Hello from a function")
my_function()
```

Creating a Function

In Python a function is defined using the **def** keyword:

```
def my_function():
    print("Hello from a function")
```

પાલાંનામાં **def** ક્રીડકની ઓપરેટોર કરેની ક્રિકતાનાંની આપણાનિંદા ક્રોલાઈંગ વધું છે.

Arguments

→ Information can be passed into function as arguments. [જુદેચી એફીની રીતે માંગીની બેનો કેવી રીતોએ હોય છે.]

→ Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

[કોઈ ની જીએ, ક્રમાંકની નામ પછી એવીલીની ઉદ્દેશ્ય કરીની જરૂરી વાર્તા હોય છે. એવી એવી ટૈપ્પણી એવીલી ઉદ્દેશ્ય હાજરી હોય, તેથી એવીની અધ્યાધ્યાત્માની જરૂર હોય.]

→ The following example has a function with one argument (fname). When the function is called, we pass along a first name, which is used inside the function to print the full name:

[નીચીના ડેટાખાળીમાં જેંક એફીલ (fname) આધ્યાત્મ કર્યું હતું હોય. જરૂરી ક્રમાંકની વ્યોદાયાંની જરૂરી હોય, એરે એવી ખરાં નાસોં આવી ખરાં કરીની હોય, જેની પ્રોફીલ પૂર્ણ એ નામ કર્યા હોય આદે ક્રમાંકની જીએ ખાલ હોય.]

```

def my_function(fname):
    print(fname + "deep")

my_function("Email")
my_function("Linus")

```

મારો નામ ક્રિકેટરનાનાં એળીં એવી છે કે
સેટાની માટે હૈ.

* → * Parameters or Argument

The terms parameters and arguments can be used for the same thing: Information that are passed into a function.

[પરમિટરમાંથી વાંચેલી એળીં કોઈ જાતીની ઉચ્ચારા વિષય
જે કેરૂમાં બધા હાંચે હૈ: આહુલી કે કે ફોનશાનમાં પરાયા
થાય હૈ.]

→ From a Function perspective

A parameter is the variable listed inside the parentheses in the function definition.

[Parameters හි තිස්සන් වූ සේවක යුතු කළ ඇති.]

An argument is the value that is sent to the function when it is called.

[ඉලිඳුවෙහි ඇතුළත් සීමැත්තු තුළු පිළිබඳ නොවා ඇති මූලික පිරි.]

* Number of Arguments

By default, a function must be called with the correct number of arguments. meaning that if your function expects 2 arguments, you have to call the function with 2 arguments, not more, and not less.

මුළුවෙන් එහි, ක්‍රියාත්මක උග්‍රහීයාවෙන් යායි
ක්‍රියාත්මක ප්‍රාග්ධන හි. ප්‍රාග්ධන හි ක්‍රියාත්මක 2 උග්‍රහීයාවෙන්
යායි යායි හි, ති තායි ක්‍රියාත්මක 2 උග්‍රහීයාවෙන් යායි නො නොවා
පුදු බඟ්‍රි, දෙශ මැටි යායි යීමා මැටි.

een GEERTjeaii 2 Ecclitoi विष्णु २०१६, एप्रील २०१६
सिंह द.

def my_function(fname, lname):

 print fname + " " + lname

my_function("Emil", "deep")

* → * Arbitrary Arguments, * args

IF you do not know how many arguments
that will be passed into your function, add a
* before the parameters name in the function
definition.

[कि तरीने जाहें आयी है तबाह किसी भावामें किटली उलीतों
जाहें कुछामें आयी, * तो किसी उकिनीशानमें रुकामीज़े
नामानी बहिली उमीदी।]

[यहा यीते किसानी दृग्गी बद्दी उलीतों चाहत रही, फिर
ही अपने घायाघाम स्थितिस्थित करी उत्तरी।]

This way the function will receive a tuple of arguments, and can access the items accordingly.

Example:

If the number of arguments is unknown, add `*` before the parameters name.
C या एकीफल वर्तमान बिंदु, * या परिसंग वाचनी निहारा जाती.

```
def my_function(*kind):
```

```
    print("The youngest child is " + kind[2])
```

```
my_function("Emilia", "tobias", "timus")
```

* →

keyword Argument

You can also send arguments with the key-value syntax. (यांत्री की = अबू दिलाला साही = दीपि नाही शिंकली असू शे.)

This way to the order of the argument does not matter. (यांत्री नित उलीलाला फाली किंवा असू असली नाही.)

```
def my_key(c11, c12, c13):
    print("my name is" + c13)
```

```
my_key(c11="deep", c12="vasu", c13="lalit")
```

* → Arbitrary keyword Arguments [**kwargs]

If you do not know how many keyword arguments that will be passed into your function, add to asterisk: ** before the parameter name in the function definition.

C ની ડાર્ની અધ્યાત્મ એ હોલ કુંઠાણ તમારા ફૂકણાનાં કેવી ક્રિએટ એલીલ જ્ઞાનાં વાપરી, તીવ્ચી ક્રિએટ: ** ફૂકણાનાં દીવાળીને નામ રાખીનાં.)

This way the function will receive a dictionary of Arguments, and can access the items accordingly.

C એવા રીતે ફૂકણાન એલીલાની જાહેર કિંદા માનીની, એવી હી મુજબ એસ્ટ્રેચિયાની વંગકણીસ અદી હાંક હૈ.)

Example :

जी कीपट एलीलो नी वांग्या आजावा द्युवा, ** ती चियामीऱ्या
तारोनी नदिला कृष्ण अंकिती:

```
def my_kwargs (** kid):  
    print ("His last name is " + kid["name"])  
  
my_kwargs (fname="deep", lname="maan")
```

* TYPE OF FUNCTION IN python

- 1). Normal Function
- 2). With Argument
- 3). Default Argument
- 4). Arbitrary Function
- 5). Arbitrary keyword Argument Function (Kwargs)
- 6). List Function
- 7). Empty Function

~~~~~ (1) Normal Function:

```
def my_function():
    print("Hello World")
```

→ Python නි [def] කිරීම සංක්‍රාන්ති ප්‍රධාන කෙටුව  
යොමු කළ ඇ.

~~~~~ (2) With Argument

```
def add(x):
    return 5*x
```

```
print(add(10))
```

Rum :- 50

→ (3) Default Argument

→ default arguments in python function are those arguments that take default values if no explicit values are passed to these arguments from the function call.

ပုဂ္ဂနယ်တွင် ဒိန်လက်များ စိန်းမြေတွင် အသုတေသနများ ဖြစ်ပါသည်။ ဒိန်လက်များ မှာ မြတ်ဆုံးမှု များ ဖြစ်ပါသည်။ ဒိန်လက်များ မြတ်ဆုံးမှု များ ဖြစ်ပါသည်။

```
def add (name="proxware")  
    print ("Company name:", name)  
add()
```

→ (4) Arbitrary Function

→ Arbitrary Arguments, * args

→ if you do not know how many arguments that will be passed into your function, add * before the parameter name in the function definition.

C ने किसी भी विलेखन को बदलने की सकती है।
यहाँ, cmd, name और age के बारे में जानकारी देता है।

→ This way the function will receive
a tuple of arguments, cmd can access
the items accordingly.

C ने किसी भी विलेखन को बदलने की सकती है।
जैसे यहाँ यहाँ के लिए cmd को उपलब्ध नहीं है।

Ex:-

```
def my_function(*data):
```

```
    print("No is:", data[0])
```

```
    print("Name is:", data[1])
```

```
    print("Id is:", data[2])
```

```
    print("std is:", data[3])
```

```
my_function(1, "deep", 12, "BCA")
```

→ C55

Arbitrary keyword Arguments function (Kwargs)

→ **kwargs allows us to pass a variable number of keyword arguments to a python function. in the function, we use the double asterisk (**) before the parameter name to denote this type of argument.

C * कार्यक्रम में अनिश्चित संख्या के कार्यक्रमों की ज़रूरी आवश्यकता है। Python कार्यक्रमों की लिंग के लिए इसका उपयोग किया जाता है। अब इसका उपयोग करने के लिए क्या कोड लिखा जाएगा? यहाँ दिए गए उदाहरण को देखें।

Ex :-

```
def argument(**data)
```

```
print("This is Iam.name is :", data  
      ["name"])
```

```
argument(name1="Android", name2="python")
```

6. list Function

python list() function takes any iterable as a parameter and returns a list. In python iterable is the object you can iterate over.

python list() ક્રિયાની કોરાયાં અધિકારીનું રહિયાયા લાગે છે એ જે એવી વાત કરે છે. ચારાંથી દૂરેસીએ એવી વોર્કશેપ હૈ કે જેની તો પુનરાબિન્ડ કરી શકતા છે.

Example

```
def listargument(data):
```

```
    print("Id", data[0])
```

```
    print("NAME", data[1])
```

```
    print("city", data[2])
```

```
listargument([1, "deep", "Ahmedabad"])
```

→ (7). Empty Function

In python, to write empty functions, we use pass statement. pass is a special statement in python that does nothing. It only works as a dummy statement.

```
def function():
    pass.
```